

SPring-8 の BL15XU で取った粉末写真像 ****.img をパソコン上で ****.bmp に変換する。変換されたファイルから、市販（あるいはフリー）ソフトを使えば、適当な範囲を切り取ったり、コントラストを調整して、デモ用に使える。

パソコンでの使用を想定し、オリジナルの 4 ピクセルを 1 ピクセルに粗視化し、かつオリジナルの 2 バイトデータを 1 バイト（256 階調）に落として、出力している。その結果、オリジナルの 62.5MB のファイル容量は 7.814MB にまで小さくしている。

その際、2つの方法で階調を落としている

0 : 線形スケール $c0 \cdot I / c1$ の線形変換 (デフォルトは $c0=1, c1=256$)

1 : 対数スケール $c2 \cdot \log_{10}(I/c3)$ の対数変換 (デフォルトは $c2=53, c3=16$)

ここで、53 とか 16 の係数は必ずしも最適値とは言えないかも知れない。ユーザーが sp8img.inf で自由に設定できる。そのファイルが存在しないときはデフォルト値を採用
実行例 (下線部がキーボード入力) Ver=20100127 からは上述のように変換の際のパラメータを環境設定ファイルから読み込む。実行画面には多少の修正があり。

C:\Y work>SP8IMG.exe

```
<<<< Convert Sp8BL15XU img to bmp or csv Ver=2010/1/26 >>>>
```

```
Input file name *.img =? Ce02_20100123a.img
```

```
Coarse graining 2*2 into 1
```

```
Now loading img file. Wait about 100 sec!
```

```
0: convert img to bmp (2000*4000 pixel)
```

```
1: monitor data and convert to csv file
```

```
Select 0 or 1 =? 0
```

```
==== BMP file out =====
```

```
Input bmp file name =? Ce021g.bmp
```

```
BG is 0: white or 1:black, =? 0
```

```
Scale is 0:Linear or 1:Log =? 1
```

```
Intensity I= 1 ~ 65,536 is scaled as 53.0*log10(I/16)
```

```
Now writting bmp file. Wait about 10 sec!
```

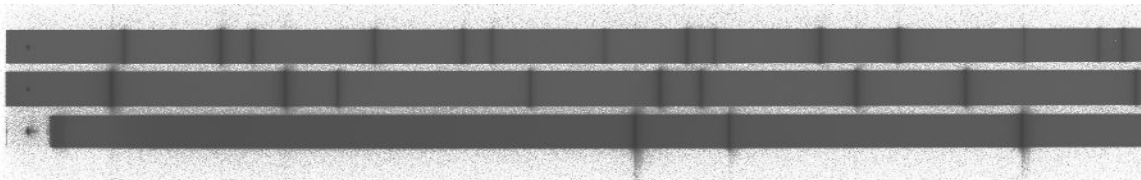
```
Output bytes=8001078
```

次に、得られた bmp ファイルの一部を切り取り、jpg に変換したものを示す。

線形スケール



対数スケール (弱い部分を強調)



Mac でも Fortran compiler を install してあれば、翻訳実行できる。

OSX でフリーのコンパイラ `gfortran` で検証済み。コンパイルしたファイルを terminal の窓で
~/> ./a.out

とすればよい。./を忘れないこと。

なお、Windows のテキストファイルを Mac で読み込んだとき、ファイル末尾の EOF コードが、`gfortran` の `compiler` ではエラーとして識別されることがあるので、Mac のテキストエディターにおいて、末尾の余計な行を消去するとよい。

また、Mac note ではプログラムの実行時間が長い場合がある。Wait about 100 sec が倍以上かかる場合もある。Fortran ではなく、C で書き直したりして、アルゴリズムの改良の余地があるようだ。

以上 2010/1/28 H.M.